(2)

ESD-TR-84-358 MTR-9480

AD-A158 599

A DESIGN EVALUATION CHECKLIST FOR USER-SYSTEM INTERFACE SOFTWARE

By

S. L. SMITH
J. N. MOSIER

OCTOBER 1984

Prepared for

DEPUTY FOR ACQUISITION LOGISTICS AND TECHNICAL OPERATIONS
ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
Hanscom Air Force Base, Massachusetts

Project No. 572A

Prepared by

THE MITRE CORPORATION
Bedford, Massachusetts
Contract No. F19628-84-C-0001

REVIEW AND APPROVAL

This techı .al report has been reviewed and is approved for publication.

N. Ann Kuo

N. ANN KUO, 2Lt, USAF
Project Manager,
Requirements Analysis

William J. Letendre

WILLIAM J. LETENDRE
Program Manager,
Computer Resource Management
Technology

FOR THE COMMANDER

Robert J. Kent

ROBERT J. KENT
Director, Computer Systems Engineering
Deputy for Acquisition Logistics
and Technical Operations

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE

AD-A158599

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION
Unclassified

1b. RESTRICTIVE MARKINGS
None

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION/DOWNGRADING SCHEDULE

3. DISTRIBUTION/AVAILABILITY OF REPORT
Approved for public release;
distribution unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
MTR-9480
ESD-TR-84-358

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION
The MITRE Corporation

6b. OFFICE SYMBOL *(If applicable)*

7a. NAME OF MONITORING ORGANIZATION

6c. ADDRESS *(City, State and ZIP Code)*
Burlington Road
Bedford, MA 01731

7b. ADDRESS *(City, State and ZIP Code)*

8a. NAME OF FUNDING/SPONSORING ORGANIZATION Deputy for Acquisition Logistics and (over)

8b. OFFICE SYMBOL *(If applicable)*
ALSE

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
F19628-84-C-0001

8c. ADDRESS *(City, State and ZIP Code)*
Electronic Systems Division
Hanscom AFB, MA 01731-5000

10. SOURCE OF FUNDING NOS.

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.
	572A		

11. TITLE *(Include Security Classification)*
A Design Evaluation Checklist for (over)

12. PERSONAL AUTHOR(S)
Sidney L. Smith, Jane N. Mosier

13a. TYPE OF REPORT
Final

13b. TIME COVERED
FROM ________ TO ________

14. DATE OF REPORT *(Yr., Mo., Day)*
1984 October

15. PAGE COUNT
66

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB. GR.

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
Computer-Based Systems, Information Systems,
Design Guidelines, ~~User System Interface~~ (over)
Human factors

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

This report proposes a checklist for evaluating the design of user-system interface (USI) software. That checklist is based on recently published ESD/MITRE guidelines for USI software design, and can be used effectively only in conjunction with the source guidelines material. The checklist items cover six general functional areas of USI design: data entry, data display, sequence control, user guidance, data transmission, and data protection.

For application to any particular system, the importance of each checklist item must be weighted in order to tailor the checklist to specific design requirements for USI software. This tailoring process should involve the joint efforts of system development managers, the software designers, and the intended system users.

(over)

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT
UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS ☐

21. ABSTRACT SECURITY CLASSIFICATION
Unclassified

22a. NAME OF RESPONSIBLE INDIVIDUAL
Claudette G. Hanley

22b. TELEPHONE NUMBER *(Include Area Code)*
(617) 271-7996

22c. OFFICE SYMBOL

DD FORM 1473, 83 APR

EDITION OF 1 JAN 73 IS OBSOLETE.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE

8a. Technical Operations

11. User-System Interface Software

18. Human Factors

19. Abstract Continued

Once design requirements have been established in this way, then any proposed or implemented USI design can be evaluated in terms of its judged compliance with the specified checklist items. Overall estimates of design quality can be derived, along with more detailed analysis of design strengths and weaknesses, in order to recommend needed design improvements.

This design evaluation checklist should prove a useful tool in Air Force system acquisition. As the USI design guidelines are revised and expanded, the checklist can be updated correspondingly. In the future, the checklist might be implemented as an on-line computer tool, with automated aids to facilitate its use. This report describes some of the potential computer aids that could be provided. keywords:

FLD 18

ACKNOWLEDGMENTS

This report was prepared by The MITRE Corporation, under MITRE Project 5720, Contract No. F19628-84-C-0001. The work reported here was sponsored by the Computer Engineering Applications Division, Deputy for Acquisition Logistics and Technical Operations of the Electronic Systems Division (ESD) of the United States Air Force Systems Command, Hanscom Air Force Base, MA 01731.

Funding for the prior development of USI design guidelines, under MITRE Project 5220, was provided by the Air Force Computer Resource Management Technology Program, Program Element 64740F. That Program supports development and transition into active use of tools and techniques needed to cope with the explosive growth in Air Force systems that use computer resources. The objectives of that Program are:

to provide for the transition to Air Force systems of computer system developments in laboratories, industry, and academia;

to develop and apply software acquisition management techniques to reduce life cycle costs;

to provide improved software design tools;

to address problems associated with computer security;

to develop advanced software engineering tools, techniques, and systems;

to support the implementation of high-order programming languages, e.g., Ada;

to improve human engineering of computer systems; and

to develop and apply computer simulation techniques in support of system acquisition.

TABLE OF CONTENTS

SECTION 1

INTRODUCTION

This report considers ways to evaluate the design of software supporting the user-system interface (USI) to computer-based information systems.

Section 1 discusses USI software generally, and the current MITRE development of guidelines for USI software design, sponsored by the Air Force Electronic Systems Division (ESD). Alternative approaches to USI software evaluation are considered, including the possibility of evaluating software by comparison with design guidelines.

Section 2 proposes a design evaluation checklist derived from published guidelines. The checklist is formatted to permit differential weighting and rating of various itemized design features. The checklist itself is presented in Appendix A, and a sample application of the checklist is illustrated in Appendix B.

Section 3 describes potential use of the design evaluation checklist if it were implemented as an on-line computer tool, with aids to facilitate review and tailoring of guidelines material, and to support the entry, analysis, and reporting of USI software design evaluation. A sample design analysis that might be generated by such a computer tool is illustrated in Appendix C.

USI SOFTWARE

What is the user-system interface? Here the phrase "user-system interface" is defined broadly to include all aspects of design that affect a system user's participation in data handling transactions. The word "user" is intended to include managers and maintainers of a system as well as its operators.

Directly observable aspects of the user interface include the physical work environment and the hardware facilities at a user's work station. Such physical features, sometimes called the man-machine interface, have been the subject of conventional human engineering study. There the concern is for illumination, seating, workplace arrangement, keyboard layout, display contrast, symbol size, etc.

Good design of the physical workplace is important, of course, but by itself is not sufficient to ensure effective job performance. Also important are less tangible aspects of information system design -- the ways in which data are stored and manipulated, including paper files and forms, if any, and the procedures and processing logic that govern data handling transactions.

Forms, procedures, and logic involve software design, the design of computer programs to permit hardware and paper to be used in conjunction with automated data processing. Effective design of USI <u>software</u> is critical to system performance.

In a recent survey, people involved in design of information systems were asked to estimate the percent of operational software devoted to implementing the user interface. Overall, the average estimate was that USI design comprises 30-35 percent of operational software. Estimates for individual systems ranged from 3 to 100 percent (Smith and Mosier, 1984a).

Because USI software is important, and also expensive, we must find ways to improve its design. For system designers, guidelines have been proposed to help define requirements for the design of user interface software. For system users, and for their agents such as ESD who sponsor system development, we need corresponding tools for evaluating USI software design.

USI DESIGN GUIDELINES

Over the past several years, there has been a continuing effort at ESD/MITRE, under Project 522A, to compile guidelines for the design of user interface software. In that effort, we have tried to distill the current wisdom of USI design experts in order to provide comprehensive advice to designers.

Each year our published compilation of guidelines has grown larger. The most recent report on this subject proposes some 679 guidelines in 450 pages, covering the general functional areas of data entry, data display, sequence control, user guidance, data transmission, and data protection (Smith and Mosier, 1984b).

Those guidelines will undergo further review and revision. An improved version of the USI design guidelines will be published in 1985. Meanwhile, the current report can provide a good starting point for deriving specific rules to govern user interface design in any particular system development program.

To help assure broad applicability USI design guidelines are expressed in general terms, which may then be amplified by specific commentary and examples. A sample of general guidelines pertaining to data entry functions is presented in Appendix A.

Topical organization of the guidelines corresponds to defined functional USI capabilities. Thus it may be possible in some degree to tailor guidelines to USI requirements. If a function is not required for a particular system, then design guidelines related to that function will not be applicable.

In many instances, software designers must interpret the guidelines based on task analysis, anticipated data processing capabilities and user skills, in order to meet the needs of a particular system. In effect, designers must translate or convert guidelines into specific design rules.

The establishment of rules for user interface design will be of significant value in system development, and should be specifically included in a design contractor's statement of work. The contractor should be required to establish design rules in advance of USI software implementation. Contractor documentation of USI design rules, and any subsequent revisions, should be maintained thereafter on a current basis to aid design coordination throughout system development (Smith, 1984).

It must be recognized that establishment of agreed rules will not in itself assure effective user interface design. No matter how clearly they are stated, those rules will still be subject to varying interpretation by software designers. Moreover, it is inevitable that some design trade-offs will be required in the practical application of different rules.

But the process of establishing and documenting the rules should ensure that serious consideration is given to this aspect of software design. And the existence of agreed rules should help focus attention and clarify discussion of user interface design issues among the sponsors, designers and ultimate users of a system.

USI DESIGN EVALUATION

Design guidelines provide a tool that can be applied by system contractors, i.e., by people responsible for conceiving and implementing USI design. Some corresponding tool should be devised for people who are responsible for evaluating USI design. Those people may be the system users, or they may represent contracting agencies such as ESD who act on behalf of the eventual system users.

Effective design evaluation requires emphasis throughout the course of system development. Evaluation must begin with timely review of design proposals in advance of implementation, continue with review and assessment of design prototypes, through operational testing of the fully designed system.

Alternative Approaches

There are several possible approaches to design evaluation, including analytic modeling, performance testing, comparison with design guidelines, review of specifications and design proposals, user surveys, etc. All of these approaches can play a useful role. The first three deserve further comment.

Analysis and Modeling

Where we seek insight into the processes that underlie information handling tasks, detailed analysis of the user actions that are required to perform various tasks permits the creation of models to predict the effects of system design on operational performance. Task modeling can help assess design differences among different systems, or to predict the effects of design changes to a particular system. Accuracy of prediction will depend upon the validity of the models, which is subject to experimental testing in the laboratory and operational testing in actual system use. The results of such analysis and modeling must be translated into specific design recommendations, which will require considerable judgment.

Performance Testing

In applications where operational performance is critical, the most direct means of evaluation is to test the performance of system users on various characteristic ("benchmark") information handling tasks. A range of users and tasks must be tested to represent fairly the actual system. Performance testing must wait for completion of system design (or at least a working prototype), or else must be based on detailed modeling if done in advance of system design. Such testing requires a considerable investment of time and effort. The results of controlled testing should provide a good prediction of actual system performance, but will require careful analysis and judgment in order to specify needed design improvements.

Comparison with Design Guidelines

Where specific design recommendations are needed, the quickest way to evaluate the user interface may be to examine design features

provided in a system (or proposed features for a system still under development) and compare those features with established design guidelines. Establishment of agreed guidelines will require the collective judgment of the sponsors (procuring agency, marketing department, etc.), designers, and prospective users of the system. Further judgment will be needed to assess design compliance with agreed guidelines. The validity of this approach to design evaluation will depend upon the good judgment of the people involved in compiling and applying the guidelines. Validity might be measured by performance testing, although that process will prove impractical as the number of design features considered (and their possible interactions) grows very large.

Comparative Advantages

These three approaches to design evaluation are complementary. A program of analysis and modeling will require performance testing for its validation. Performance testing must in turn be analyzed to interpret its results. And both approaches will benefit from reference to design guidelines in order to generate specific design recommendations. For effective system development and evaluation, all three approaches may be needed. The comparative strengths and weaknesses of different approaches are summarized here:

	Approaches to Evaluation		
Goals of Evaluation	Analysis, Modeling	Performance Testing	Design Guidelines
Insight into task processes	Good	Fair	Poor
Performance prediction	Fair	Good	Poor
Specific design recommendations	Fair	Fair	Good
Ease and speed of evaluation	Fair	Poor	Good
Evaluation early in system development	Fair	Poor	Good
User involvement in design decisions	Poor	Fair	Good

Problems of Software Evaluation

System evaluation must cover both hardware and software design. As a practical matter, the evaluation of proposed hardware may be relatively straightforward. User interface hardware can be checked for compliance with the provisions of current DoD human engineering standards for equipment design, MIL-STD-1472, as well as with system specifications. Some aspects of hardware design can be assessed simply by observation. Are the corners of equipment enclosures rounded rather than sharp? Just look at the blueprints, or catalogs, or actual equipment, and see.

By contrast, evaluating user interface software will be more difficult, particularly in advance of design implementation. For many ESD system acquisitions, it is not possible to buy off-the-shelf software, out of a catalog. Instead, user interface software must often be designed anew for each system application. And the "blueprints" for a proposed new software design may be hard to read and evaluate.

Evaluation of USI software may pose difficult problems even after its implementation. It may be necessary to operate the system under controlled conditions to confirm that design objectives have been achieved.

Are data displays formatted consistently from one transaction to another? We may need to generate different displays and compare them. To facilitate that comparison, we may need to request hard-copy printouts of various displays.

Are error messages worded clearly and consistently? That may be harder to assess. We cannot expect an evaluator to make every conceivable type of error while operating a system, trying to generate a full range of error messages. Instead, we may need to examine a detailed software design specification in which all error messages have been spelled out.

What other features should we examine in evaluating USI software design? The range will extend far beyond considerations of display formatting and the wording of error messages. We need some comprehensive listing of all significant features that should be considered.

Specificity is important here. For design evaluation to be useful, it must produce detailed recommendations, outlining for system designers exactly which features need improvement. It is not enough to shake our heads and say that a user interface looks clumsy

or confusing, that a system is difficult to learn or difficult to use. We must try to provide a more specific diagnosis of design deficiencies.

Tools for Software Evaluation

What tools can we devise for evaluating USI software design? It would seem sensible to start with the published design guidelines. But there the sheer bulk of the published guidelines material poses a serious problem.

We cannot expect an evaluator to read through a book containing hundreds of guidelines, checking each one for design compliance and annotating design discrepancies in the margins. An evaluator will need to read the guidelines once, during initial assimilation of the material, but surely would not wish to do so repetitively when assessing the user interface for different systems, or for one system at different stages.

If a design evaluation were actually made by checking and annotating a large book of guidelines, there would still remain the difficult task of pulling together those notes to create a coherent overall assessment of the strengths and weaknesses of an evaluated user interface design.

Certainly the published guidelines can serve as a starting point for USI design evaluation. But it is apparent that some more compact format will be needed for recording and reporting the results of design evaluation. In the next section, a checklist format is proposed for that purpose.

SECTION 2

CHECKLIST FOR DESIGN EVALUATION

In order to create a compact design evaluation record, we must convert the fully-worded guidelines into a more condensed listing. In such a list, each itemized design feature could be checked for compliance and discrepancies noted. The resulting checklist format should prove relatively easy to use -- for the initial recording of evaluative judgments, and for the subsequent analysis and reporting of design evaluation. That checklist is presented in Appendix A.

The use of checklists for design evaluation is, of course, not a new idea. A recent AMRL report recommends checklists among other means of assessing compliance with human engineering standards (Geer, 1981, pp. 156-159). Most current checklists are intended to evaluate hardware rather than software, but checklists have also been proposed for software evaluation (Cordes, 1980). The general approach, in terms of checklist formatting and use, should prove much the same for evaluating USI software design.

The checklist presented here is based on our most recent compilation of design guidelines for USI software (Smith and Mosier, 1984b). Each of the 679 guidelines is represented in the checklist by its title. For a person familiar with the guidelines, those titles will serve as reminders of the guidelines material. First-time checklist users, however, will need to refer frequently to the source guidelines in order to determine exactly what design requirements are specified there.

The checklist proposed here is simple in concept. Each design guideline is represented as an item entry in the list. After each item is a space to indicate an (optional) weighting of importance. Next there is space to record an estimate of compliance. Finally there is space in which to note whether comments have been added to cite specific design deficiencies or to provide other kinds of information.

A sample showing how the checklist might be used, with imaginary entries for purposes of illustration, is shown in Appendix B. That sample shows just a portion of a complete USI design evaluation checklist, the portion that corresponds with guidelines for data entry functions.

Several aspects of checklist use -- functional coverage, weighting importance, rating compliance, and annotation -- all deserve further discussion.

FUNCTIONAL COVERAGE

A user of the checklist will note significant differences in coverage for different USI functions. Some functions are thoroughly covered, but some are not. Under data entry, for example, there are 24 items dealing with position designation, but only two for direction designation. Graphic functions are scarcely covered at all in the present checklist.

This differential coverage results, of course, from differences in coverage within the source guidelines material from which the checklist was derived. In the compilation of design guidelines there are some areas in which our knowledge is greater than in others, and thus more information can be offered to interface designers. This condition of differential knowledge will persist for some years to come, and so the problem of differential coverage in the design evaluation checklist will persist also.

One consequence is that design evaluation may be biased toward those functions that are covered well in the checklist, and important functions with little coverage may be neglected. How great is this problem? Until we have gained experience in using the design evaluation checklist, we cannot answer that question.

How can a person responsible for design evaluation cope with this problem? Several corrective measures are available. The evaluator might propose additional items for checklist functions where coverage seems inadequate, thus strengthening weak portions of the evaluation tool. For example, if a particular system has an extensive need for graphic interaction, then it may be possible to establish agreed design rules for graphic functions and represent those rules by items added to the present checklist.

As an alternative or supplementary approach, an evaluator might assign exceptionally high weights to a few listed items, where only those few deal with an important user interface function. The process of differential weighting of items is discussed later in this section.

Another problem of checklist coverage is specificity. If design guidelines are to be applicable to a variety of system applications, they must be stated generally. This will also be true of a checklist derived from those guidelines.

For any particular system application, it will often be necessary to tailor the general guidelines to become system-specific design rules. It is only such specific rules that will provide effective guidance to designers.

Where guidelines are revised to achieve specificity, then it may be necessary to revise corresponding checklist items as well. Perhaps the titles of checklist items need not be changed, but those titles should be interpreted as referring to whatever system-specific design rules have been adopted, rather than to the general guidelines that underlie those rules.

WEIGHTING IMPORTANCE

In the design evaluation checklist there will be hundreds of items to be considered. For any particular system design, some of those items may not be relevant. For example, if a decision is made not to provide multicolor displays, then any guidelines pertaining to color coding will not be applicable to that system.

There are two ways to deal with this situation. One way would be to tailor the design evaluation checklist so that it includes only those items relevant to a particular system. That tailoring could be difficult for a paper checklist, where many pages might have to be retyped. Moreover, it might prove disconcerting for an experienced evaluator to find the contents of the checklist changing for each different system evaluation.

A simpler solution is to preserve the checklist in standard format, consistent in contents and ordering, from one system to another. An experienced evaluator will find each item where it is expected, always in the same place. Items not applicable to a particular system might be marked "N/A" in the course of design evaluation, and could be ignored as long as that designation is retained.

Just which items are not applicable can presumably be decided by users (and/or sponsors) in negotiation with the system design contractor, or perhaps with a USI software subcontractor, in order to help define requirements before USI software is designed. And that decision might be reviewed periodically during successive stages of system design, along with more general review of required USI functional capabilities.

Eliminating from consideration any checklist items that have been judged not applicable to a particular system design, one may question whether the remaining items are all equally important. An evaluator might well conclude that some of those items are more important than others, and wish to assign differential weightings to discriminate among the various items.

Such weightings could be simple categorical judgments among just a few designations -- perhaps "Required" or "Desirable" in addition to "Not Applicable". Or weighting judgments could be represented along a numeric scale -- perhaps zero to ten, where zero denotes items judged to be not applicable, and ten denotes items judged to be of critical importance. (Such numeric weightings are illustrated by the sample in Appendix B.) Numeric weightings would offer advantages for subsequent computational analysis of user interface design evaluation, as discussed in Section 3.

In practice, the assignment of differential weightings to checklist items must necessarily be judgmental. For any particular item, different weightings might be assigned by system users, by software designers, or by their human engineering consultants.

The checklist can help focus attention on design decisions, but cannot tell people how to make those decisions. Design biases may be introduced in the weighting process. Users might weight heavily those design features that have been missing in the past, hoping thereby to encourage their inclusion in a new system. Designers might weight heavily those features that they believe will be easy to provide.

Whose judgment should prevail? The best solution is to ensure that weightings will not be established by only one person. All groups concerned with system development should have an opportunity to participate in the weighting process.

RATING COMPLIANCE

In using the checklist, an evaluator must judge for each item the degree of compliance in a proposed or implemented user interface design. Different rating schemes might be adopted to record the evaluator's judgment.

One might adopt a simple YES/NO rating to indicate judged design compliance for each checklist item. At least that seems simple. But the evaluator will still have to exercise judgment as to whether the observed degree of design compliance is substantial enough to justify a YES rating for any particular item. The rating itself would not reflect that judgment process. We would not know from the rating whether the evaluator's YES was a confident assessment of full compliance or was assigned in grudging acknowledgment of marginal compliance.

Considering the general process and purpose of USI design evaluation, it will probably prove more useful if the evaluator's

judgment is reflected more directly in the ratings assigned. The sample checklist application in Appendix B employs numeric ratings that range from zero (no compliance) to ten (full compliance). With such numeric ratings, varying degrees of marginal compliance can be acknowledged in the design evaluation record, as a means of pointing the way toward needed design improvements.

Although numeric ratings may permit an evaluator to convey design strengths and weaknesses more effectively, we must recognize that numeric evaluation can be misleading. Qualitative ratings such as YES/NO, or GOOD/FAIR/POOR, are clearly judgmental. Numeric ratings, however, may imply a degree of objectivity that is not justified by the real nature of the design evaluation process. Thus evaluators must be conservative when reporting and interpreting the results of numeric rating.

As a practical matter, rating compliance with agreed design guidelines will involve a great many other problems, of the kind discussed in the previous section of this report. Particularly in early stages of system development, when ratings can be made only in terms of specifications and other documentation of proposed designs, it may not be possible to determine the degree of compliance with various specific design requirements. For some items in the checklist, compliance ratings may have to be marked as unknown.

Even for a fully documented and completed system, it is obvious that evaluation of compliance with design objectives must be in large degree judgmental. And again we must consider the question of whose judgment should be sought.

Designers may wish to apply the checklist to their own work, perhaps as a periodic measure to determine interim success in meeting objectives. Any formal evaluation, however, should involve system users and their representatives. They may choose to form their judgments in consultation with designers, but they should record and report their design evaluation independently, in their own behalf.

ANNOTATION

Weightings and ratings will permit a quantitative assessment of USI design, but in many cases will not convey the exact nature of deficiencies observed during design evaluation. It will help in prescribing improvements to USI design, if the evaluator can note specific examples of design deficiencies for low-rated items in the checklist.

Such annotation would take the form of appended comments. There is not sufficient space in the checklist itself to provide a full description of observed deficiencies. It is proposed instead that the checklist simply include for any item an indication that a comment has been appended. That indication might be a check or other special mark following the assigned rating. Appended comments would be labeled with the number codes of corresponding checklist items.

Evaluator comments will not always describe design deficiencies, although that would be their primary function. Some comments might raise questions about design features considered doubtful by the evaluator. Other comments might commend observed examples of good design practice.

Some comments might pertain to the weightings assigned to different checklist items, dealing with design requirements rather than design achievement. For example, there might be comments explaining why particular items have been assigned zero weight. Although such comments would not affect design evaluation, they could be useful if design requirements must later be reconsidered.

Finally, some comments might pertain to functional coverage, perhaps indicating a special interpretation for an item in the checklist, or the conversion of a generally stated item into a more specific design rule.

PRACTICAL APPLICATION

Effective application of the design evaluation checklist will require that user interface design be specified, implemented, and/or documented in sufficient detail to permit review in terms of features specified in the checklist. Otherwise, many of the checklist items can be rated only with question marks. In such a situation, application of the checklist can merely indicate inadequacy of design documentation rather than inadequacy of the design itself.

In its initial applications, we may expect that the design evaluation checklist will be used solely for guidance rather than for contractual enforcement. Certainly if a contractor has not been required to establish and follow USI design guidelines in the first place, then it would not be reasonable to impose retroactively some set of design evaluation criteria derived from guidelines. If the design evaluation checklist proves useful in its initial applications, however, we may see its future adoption as a more formal contractual instrument.

SECTION 3

POTENTIAL COMPUTER AIDS

The design evaluation checklist proposed here should prove a useful tool. Looking ahead, however, we can envision ways in which this tool could be made even more effective. A promising means for facilitating its use would be to implement the checklist as an on-line computer tool. Computer aids could be provided for making entries to the checklist, for reviewing and tailoring associated guidelines material, and for the analysis and reporting of design evaluation. These topics are discussed here.

CHECKLIST ENTRY AND REVIEW

At the simplest level, a checklist maintained on-line in a computer could help the entry and editing of evaluator judgments -- the weightings of relative item importance, the ratings of judged design compliance, and any associated annotation including evaluator comments.

Checklist entries will reflect evaluator judgment, and judgments may change from time to time. Changes to paper forms may be messy to make, and/or involve tedious transcription of records.

Moreover, the process of design evaluation will often be continuous, repeated at different stages of system development. Successive evaluations will generally indicate design improvements. Ratings for many items will stay the same, but ratings for some items may be revised upward, and further comments added to the evaluation record. Such successive changes would be easier to make in a computer-stored checklist than in a paper record.

Review of checklist entries would also be easier by accessing computer-stored records. With computer aids, an evaluator could request direct display of any portion of the checklist, in any of its successively applied versions, without having to leaf through a growing sheaf of paper records. Computer aids would be especially helpful for displaying checklist annotation, permitting an evaluator to review directly the comments associated with any selected item, rather than having to search separate sheets of paper.

Which kind of checklist will prove preferable -- the paper version or the on-line computer aid -- will depend upon conditions of use. For field use, where an evaluator must observe a system under test, the paper checklist would seem more convenient. For

desk use, when an evaluator is reviewing design specifications, display printouts, etc., an automated checklist could offer worthwhile advantages. That automated checklist could be used to help transcribe ratings from paper checklist records taken in the field.

REVIEWING GUIDELINES MATERIAL

One significant feature of the checklist format is that each item expresses a design guideline in condensed wording, with no full statement of the guideline nor any explanatory examples and comment. An evaluator who is not familiar with the guidelines material will have to refer back and forth between checklist titles and their corresponding source guidelines. In effect, an evaluator using a paper checklist must keep at hand not only the checklist itself but also a book of design guidelines.

A better alternative would be to reconfigure the paper checklist to become an on-line checklist supported by automated computer aids, including computer storage and display of associated guidelines material. Using such an automated checklist, an evaluator might query any particular item to request immediate computer display of explanatory material equivalent to that available in the published guidelines.

TAILORING GUIDELINES

An automated checklist could be of even greater value for design evaluation in applications where guidelines have been tailored to become system-specific rules. In such a case, published guidelines would not provide a complete account of the desired design features. An evaluator would have to refer also to whatever additional material may have been approved for design guidance.

Computer aids could help ensure reliable reference to agreed system-specific design rules, in effect displaying tailored rather than standard guidelines material. With such an automated checklist, access to explanatory material would certainly be easier. The evaluator would not need a book of design guidelines, plus whatever further notes pertain to system-specific tailoring. Instead, the evaluator <u>would</u> need a suitable computer terminal.

Once guidelines had been tailored with computer aids, computer facilities could also be used to print out a tailored version of the design evaluation checklist. That revised checklist could then be used in field testing with the same convenience as the "standard" checklist presented here.

DESIGN ANALYSIS

Where an automated checklist could really help an evaluator is in analyzing and summarizing the ratings that have been made, particularly if differential weightings have been assigned to the various items. Computer aids for analysis could automatically multiply ratings of design compliance by appropriate weightings, sum the results to derive an overall assessment of USI design quality, and display the results in different ways to provide more detailed diagnostic information.

Computer analysis could provide a diagnostic profile assessing design quality for the different functional areas supported by user interface software, for identified functions within those areas, and even for specific highly-weighted items pertaining to particular functions. A sample of what might be obtained from a computer-generated design analysis is illustrated in Appendix C. Such diagnostic evaluation can be tedious to compute manually, but would be quite easy to generate with appropriate computer aids.

FUTURE WORK

For use at ESD/MITRE, a logical next step will be to reconfigure the checklist as an automated tool, so that it can be supported with on-line computer aids of the kind suggested here -- aids for explaining checklist items, for assigning weightings and ratings, and for the diagnostic analysis and reporting of design evaluation results. An on-line interface to such an automated checklist will be designed in FY85 under MITRE Project 5720. Future implementation and testing of an automated design evaluation checklist will require further work.

Once implemented, an automated checklist would offer potential benefits beyond its immediate use for USI design evaluation. An automated checklist would provide an example of how computer tools might be applied more generally to improve corporate effectiveness in system engineering for acquisition programs. We can anticipate that such computer aids for focusing expert knowledge in support of our technical work will contribute strongly to Air Force system acquisition in the decades ahead.

REFERENCES

Cordes, R. E. (1980). Software-user interface evaluation: Methodology and tools. In Proceedings of the Human Factors Society 24th Annual Meeting (pp. 395-399). Santa Monica, CA: Human Factors Society.

Geer, C. W. (1981, September). Human Engineering Procedures Guide (Technical Report AFAMRL-TR-81-35). Wright-Patterson Air Force Base, OH: Air Force Aerospace Medical Research Laboratory.

MIL-STD-1472C, Revised. (1983, September). Military Standard: Human Engineering Design Criteria for Military Systems, Equipment and Facilities. Washington, DC: Department of Defense.

Smith, S. L. (1984, April). User-System Interface Design in System Acquisition (Technical Report ESD-TR-84-158). Hanscom Air Force Base, MA: USAF Electronic Systems Division. [NTIS No. AD A140 956]

Smith, S. L., and Mosier, J. N. (1984a). The user interface to computer-based information systems: A survey of current software design practice. In Proceedings of INTERACT'84 IFIP Conference on Human-Computer Interaction (pp. 2.35-2.39). Amsterdam: Elsevier.

Smith, S. L., and Mosier, J. N. (1984b, September). Design Guidelines for User-System Interface Software (Technical Report ESD-TR-84-190). Hanscom Air Force Base, MA: USAF Electronic Systems Division.

APPENDIX A

USI DESIGN EVALUATION CHECKLIST

This appendix presents the checklist for evaluating USI software design. As discussed in the text of this report, checklist coverage is governed by the source guidelines. For functions such as graphics where there are currently few guidelines, or none at all, checklist coverage is correspondingly weak.

The checklist will improve over the next several years as the guidelines themselves improve. Meanwhile, means might be found to compensate for current weaknesses in the checklist, perhaps by expanding coverage in weak areas to add specific design rules for particular system applications.

In this checklist, the item descriptor in each line simply replicates the title of the source guideline. Ideally, all of those titles should be informative, adequate in themselves to convey the intent of the source guidelines. But that ideal is difficult to achieve. Users of this checklist must refer to source guidelines in ESD-TR-84-190 (Smith and Mosier, 1984b) to ensure that design requirements are clearly understood for each item.

Within the checklist, items are numbered sequentially for each USI function in order to permit convenient referencing. Within each function, there is coverage of various subordinate topics indicated by the item titles. Each item that introduces a new topic is marked with a black dot (•) before its title. When succeeding items deal with the same topic, each is marked with a white dot (o) to imply that it should be interpreted in conjunction with other related items.

In practical field use, the design evaluation checklist would differ somewhat in format from that shown here. There should probably be a cover sheet with instructions explaining use of the checklist. There might also be spare sheets, or perhaps blank items inserted in the checklist, to permit recording desired USI software features that are not represented in the checklist, and spare sheets for appended comments. Those are omitted here in the interests of compactness.

Readers who wish to use the design evaluation checklist are invited to contact the authors of this report in order to exchange information concerning the details of its practical application. By sharing our experience with this new tool, we shall all learn more quickly how it may best be used.

DESIGN EVALUATION FOR USI SOFTWARE

System: ____________
Date: ____________
Evaluator: ____________

DATA ENTRY — General 1.0	Guideline Number	Weight	Rating	Comment
• Entry via primary display	-1	___	___	.
• Feedback during data entry	-2	___	___	.
o Fast response	-3	___	___	.
• Single method for entering data	-4	___	___	.
• Defined display areas for data entry	-5	___	___	.
• Consistent method for data change	-6	___	___	.
• User-paced data entry	-7	___	___	.
• Explicit ENTER action	-8	___	___	.
o ENTER key labeling	-9	___	___	.
• Explicit CANCEL action	-10	___	___	.
• Feedback for completion of data entry	-11	___	___	.
o Feedback for repetitive data entries	-12	___	___	.
o Feedback when changing data	-13	___	___	.
• Keeping data items short	-14	___	___	.
o Partitioning long data items	-15	___	___	.
• Optional abbreviation	-16	___	___	.
o Distinctive abbreviation	-17	___	___	.
o Consistent abbreviation rule	-18	___	___	.
o Minimal exceptions to abbreviation rule	-19	___	___	.
o Minimal deviation from abbreviation rule	-20	___	___	.
o Fixed abbreviation length	-21	___	___	.
o Clarifying unrecognized abbreviations	-22	___	___	.
• Prompting data entry	-23	___	___	.
• Character entry via single keystroke	-24	___	___	.
o Minimal shift keying	-25	___	___	.
• Upper/lower case equivalent	-26	___	___	.
• Decimal point optional	-27	___	___	.
• Leading zeros optional	-28	___	___	.
• Single/multiple blanks equivalent	-29	___	___	.

			Weight	Rating	Comment
DATA ENTRY	**Position Designation**	**1.1**			
• Distinctive cursor		-1	__	__	.
o Non-obscuring cursor		-2	__	__	.
o Precise pointing		-3	__	__	.
• Explicit activation		-4	__	__	.
• Fast Response		-5	__	__	.
• Stable cursor		-6	__	__	.
• Responsive cursor control		-7	__	__	.
• Consistent incremental positioning		-8	__	__	.
o Variable step size		-9	__	__	.
o Proportional spacing		-10	__	__	.
• Continuous cursor positioning		-11	__	__	.
• Direct pointing		-12	__	__	.
o Large pointing area for option selection		-13	__	__	.
• Cursor control at keyboard		-14	__	__	.
• Compatible control of cursor movement		-15	__	__	.
• Minimal use of multiple cursors		-16	__	__	.
o Distinctive multiple cursors		-17	__	__	.
o Distinctive control of multiple cursors		-18	__	__	.
o Compatible control of multiple cursors		-19	__	__	.
• Consistent HOME position		-20	__	__	.
• Consistent cursor placement		-21	__	__	.
• Easy cursor movement to data fields		-22	__	__	.
• Display format protection		-23	__	__	.
• Data entry independent of cursor placement		-24	__	__	.

DATA ENTRY	**Direction Designation**	**1.2**			
• Analog entry of estimated direction		-1	__	__	.
• Keyed entry of quantified direction		-2	__	__	.

		Weight	Rating	Comment
DATA ENTRY — Text 1.3				
• Adequate display capacity	-1	___	___	.
• Full editing capabilities during text entry	-2	___	___	.
• Free cursor movement	-3	___	___	.
o Control entries distinct from text	-4	___	___	.
• Natural units of text	-5	___	___	.
o Control entry based on units of text	-6	___	___	.
o Highlighting specified text	-7	___	___	.
o Cursor movement by units of text	-8	___	___	.
• String search	-9	___	___	.
o Upper/lower case equivalent in search	-10	___	___	.
o Specifying case in search	-11	___	___	.
o Global search and replace	-12	___	___	.
o Case in global search and replace	-13	___	___	.
• Automatic pagination aids	-14	___	___	.
o User control of pagination	-15	___	___	.
o Controlling integrity of text units	-16	___	___	.
• Automatic line break	-17	___	___	.
o Consistent word spacing	-18	___	___	.
o Hyphenation by users	-19	___	___	.
• Format control by user	-20	___	___	.
• Establishing predefined formats	-21	___	___	.
o Storing user-defined formats	-22	___	___	.
• Moving text	-23	___	___	.
• Storing frequently used text	-24	___	___	.
• Necessary data displayed	-25	___	___	.
o Text distinct from annotation	-26	___	___	.
• Printing for proofreading	-27	___	___	.
• Text displayed as printed	-28	___	___	.
• Flexible printing options	-29	___	___	.
• Information on printing status	-30	___	___	.
• Auditory signals for alerting users	-31	___	___	.
• Protecting text during page overruns	-32	___	___	.
• Confirming actions in DELETE mode	-33	___	___	.
• Reversible actions	-34	___	___	.
• User confirmation of editing changes	-35	___	___	.

	Weight	Rating	Comment

DATA ENTRY	Data Forms 1.4			
• Single entry of related data	-1	___	___	.
• Flexible interrupt	-2	___	___	.
• Minimal use of delimiters	-3	___	___	.
o Standard delimiter character	-4	___	___	.
• Data field labels	-5	___	___	.
o Consistent labeling	-6	___	___	.
o Protected labels	-7	___	___	.
o Labels close to data fields	-8	___	___	.
• Marking field boundaries	-9	___	___	.
o Prompting field length	-10	___	___	.
o Marking required/optional data fields	-11	___	___	.
• Automatic justification of variable-length entries	-12	___	___	.
• Explicit tabbing to data fields	-13	___	___	.
• Distinctive label format	-14	___	___	.
o Consistent label format	-15	___	___	.
o Label punctuation as entry cue	-16	___	___	.
• Informative labels	-17	___	___	.
• Data format cueing in labels	-18	___	___	.
o Labeling units of measurement	-19	___	___	.
o Familiar units of measurement	-20	___	___	.
o Alternative units of measurement	-21	___	___	.
• Form compatible for data entry and display	-22	___	___	.
o Form compatible with source documents	-23	___	___	.
• Minimal Cursor positioning	-24	___	___	.
o Data items in logical order	-25	___	___	.
• Automatic cursor placement	-26	___	___	.

DATA ENTRY	Tables 1.5			
• Tables for related data sets	-1	___	___	.
• Distinctive labels	-2	___	___	.
o Informative labels	-3	___	___	.
• Tabbing within rows	-4	___	___	.
o Tabbing within columns	-5	___	___	.
• Automatic justification of entries	-6	___	___	.
o Justification of numeric entries	-7	___	___	.
• Aiding entry of duplicative data	-8	___	___	.
• Row scanning cues	-9	___	___	.

Comment
Rating
Weight

DATA ENTRY	Graphics	1.6

(No entries)

DATA ENTRY	Data Validation	1.7

		Weight	Rating	Comment
• Automatic data validation	-1	___	___	.
• Accepting correct entries	-2	___	___	.
• Non-disruptive error messages	-3	___	___	.
• Deferral of required data entry	-4	___	___	.
o Reminder of deferred entry	-5	___	___	.
• Timely validation of sequential transactions	-6	___	___	.
• Optional item-by-item validation	-7	___	___	.

DATA ENTRY	Other Data Processing	1.8

		Weight	Rating	Comment
• Default values	-1	___	___	.
o User definition of default values	-2	___	___	.
o Display of default values	-3	___	___	.
o Easy confirmation to enter default values	-4	___	___	.
o Temporary replacement of default values	-5	___	___	.
• Automatic generation of routine data	-6	___	___	.
• Automatic computation of derived data	-7	___	___	.
• User review of prior entries	-8	___	___	.
• Automatic entry of redundant data	-9	___	___	.
• Automatic cross-file updating	-10	___	___	.

DATA ENTRY	Design Change	1.9

		Weight	Rating	Comment
• Flexible design for data entry	-1	___	___	.

Comment
Rating
Weight

DATA DISPLAY	General 2.0	Weight	Rating	Comment
• Necessary data displayed	-1	___	___	.
o Only necessary data displayed	-2	___	___	.
• Data displayed in usable form	-3	___	___	.
• Data display consistent with user conventions	-4	___	___	.
o Establishing display standards	-5	___	___	.
• Consistent display format	-6	___	___	.
• User control of data display	-7	___	___	.
o User changes to displayed data	-8	___	___	.
o Protection of displayed data	-9	___	___	.
• Context for displayed data	-10	___	___	.
• Familiar wording	-11	___	___	.
o Consistent wording	-12	___	___	.
o Consistent wording across displays	-13	___	___	.
• Minimal use of abbreviation	-14	___	___	.
o Consistent abbreviation	-15	___	___	.
o Distinctive abbreviations	-16	___	___	.
o Dictionary of abbreviations	-17	___	___	.
o Minimal punctuation of abbreviations	-18	___	___	.

DATA DISPLAY	Data Type 2.1	Weight	Rating	Comment
• Appropriate data types	-1	___	___	.

Comment
Rating
Weight

DATA DISPLAY - Data Type	Text	2.1.1

		Weight	Rating	Comment
• Conventional text display	-1	___	___	.
• Consistent text format	-2	___	___	.
o Conventional use of mixed case	-3	___	___	.
o Separation of paragraphs	-4	___	___	.
o Consistent word spacing	-5	___	___	.
o Minimal hyphenation	-6	___	___	.
o Conventional punctuation	-7	___	___	.
• Clarity of wording	-8	___	___	.
o Sentences begin with main topic	-9	___	___	.
o Simple sentence structure	-10	___	___	.
o Concise wording	-11	___	___	.
o Distinct wording	-12	___	___	.
o Affirmative sentences	-13	___	___	.
o Active voice	-14	___	___	.
o Temporal sequence	-15	___	___	.
o Lists for related items	-16	___	___	.
o Single-column list format	-17	___	___	.
o Logical list ordering	-18	___	___	.
o List ordering in multiple columns	-19	___	___	.
o Hierarchic structure for long lists	-20	___	___	.
• Abbreviations defined in text	-21	___	___	.

DATA DISPLAY - Data Type	Data Forms 2.1.2	Weight	Rating	Comment
• Forms for related data	-1			
• Visually distinctive data fields	-2			
o Data field labeling	-3			
o Descriptive wording of labels	-4			
o Consistent wording of labels	-5			
o Distinctive wording of labels	-6			
o Consistent label location	-7			
o Distinctive label format	-8			
o Labels close to data fields	-9			
o Labeling units of measurement	-10			
• Consistent format across displays	-11			
o Form compatible for data entry and display	-12			
• Consistent format within data fields	-13			
• Partitioning long data items	-14			
• Distinguishing blanks from nulls	-15			

DATA DISPLAY - Data Type	Tables 2.1.3	Weight	Rating	Comment
• Tables for data comparison	-1			
• Column and row labels	-2			
o Labeling units of measurement	-3			
• Justification of numeric data	-4			
o Justification of alphabetic data	-5			
• Logical organization	-6			
o Tables referenced by first column	-7			
o Items paired for direct comparison	-8			
• Distinctive labeling	-9			
o Numbered items start with "1"	-10			
o Repeated elements in hierarchic numbering	-11			
• Row scanning cues	-12			
o Column scanning cues	-13			
o Consistent column spacing	-14			
• Consistent label format	-15			

Comment
Rating
Weight

DATA DISPLAY - Data Type	Graphics	2.1.4	Weight	Rating	Comment
• Graphic display for data comparison	-1		___	___	.
o Graphic displays for monitoring data change	-2		___	___	.
• Conventional flowchart orientation	-3		___	___	.
• Standarized graphics symbology	-4		___	___	.

DATA DISPLAY - Data Type	Combination	2.1.5	Weight	Rating	Comment
• Mixing text with figures	-1		___	___	.

DATA DISPLAY	Density	2.2	Weight	Rating	Comment
• Necessary data displayed	-1		___	___	.
o Only necessary data displayed	-2		___	___	.

DATA DISPLAY	Format	2.3	Weight	Rating	Comment
• Consistent format	-1		___	___	.
o Distinctive display elements	-2		___	___	.
• Paging crowded displays	-3		___	___	.
o Related data on same page	-4		___	___	.
o Page labeling	-5		___	___	.
• Windows for related data sets	-6		___	___	.
o Integrated display	-7		___	___	.
o Adequate window size	-8		___	___	.
• Display title at top	-9		___	___	.
o Command entry, prompts, messages at bottom	-10		___	___	.
• Logical data oganization	-11		___	___	.
o Grouping for data comparison	-12		___	___	.
o Data grouped by sequence of use	-13		___	___	.
o Data grouped by function	-14		___	___	.
o Data grouped by importance	-15		___	___	.
o Data grouped by frequency	-16		___	___	.
o Data grouped alphabetically or chronologically	-17		___	___	.

		Weight	Rating	Comment
DATA DISPLAY Coding	**2.4**			
• Highlighting critical data	-1	___	___	.
• Coding by data category	-2	___	___	.
• Meaningful codes	-3	___	___	.
o Familiar coding conventions	-4	___	___	.
• Definition of display codes	-5	___	___	.
• Consistent coding across displays	-6	___	___	.
o Alphanumeric coding	-7	___	___	.
o Consistent case in alphabetic coding	-8	___	___	.
o Combining letters and numbers	-9	___	___	.
o Short codes	-10	___	___	.
• Special symbols	-11	___	___	.
o Consistent use of special symbols	-12	___	___	.
o Markers close to items marked	-13	___	___	.
• Shape coding	-14	___	___	.
o Establishing standards for shape coding	-15	___	___	.
• Line coding	-16	___	___	.
o Underlining for emphasis	-17	___	___	.
o Coding by line length	-18	___	___	.
o Coding by line direction	-19	___	___	.
• Size coding	-20	___	___	.
o Adequate differences in size	-21	___	___	.
• Brightness coding	-22	___	___	.
o Brightness inversion	-23	___	___	.
• Color coding	-24	___	___	.
o Conservative use of color	-25	___	___	.
o Adding color to formatted displays	-26	___	___	.
o Redundant color coding	-27	___	___	.
o Unique assignment of color codes	-28	___	___	.
o Conventional assignment of color codes	-29	___	___	.
o Limited use of blue	-30	___	___	.
• Blink coding	-31	___	___	.
o Blinking marker symbols	-32	___	___	.
o Optimal blink rate	-33	___	___	.
• Coding with texture, focus, motion	-34	___	___	.
• Auditory coding	-35	___	___	.
o Distinctive auditory coding	-36	___	___	.
o Voice coding	-37	___	___	.

Comment
Rating
Weight

DATA DISPLAY	Generation 2.5	Weight	Rating	Comment
• User selection of data for display	-1	___	___	.
• Display identification labels	-2	___	___	.
o Meaningful display labels	-3	___	___	.
o Consistent format for display labels	-4	___	___	.
• Fast response to display request	-5	___	___	.
o Signaling completion of display output	-6	___	___	.
• Regenerating changed data	-7	___	___	.
o Replacing changed data	-8	___	___	.
• Printing displays locally	-9	___	___	.

DATA DISPLAY	Framing 2.6	Weight	Rating	Comment
• Integrated display	-1	___	___	.
• Easy paging	-2	___	___	.
o Continuous numbering in multipage lists	-3	___	___	.
o Labels for multipage tables	-4	___	___	.
• Annotating display of continued data	-5	___	___	.
o Numbering display pages	-6	___	___	.
• Consistent orientation for display framing	-7	___	___	.
o Windowing with free cursor movement	-8	___	___	.
• Labeling display framing functions	-9	___	___	.
o Labeling windowing functions	-10	___	___	.
o Labeling scrolling functions	-11	___	___	.

DATA DISPLAY	Update 2.7	Weight	Rating	Comment
• Automatic display update	-1	___	___	.
• Readability of changing data	-2	___	___	.
o Visual integration of changing graphics	-3	___	___	.
• Display freeze	-4	___	___	.
o Labeling display freeze	-5	___	___	.
o Signaling changes to frozen data	-6	___	___	.
o Resuming update after display freeze	-7	___	___	.

Comment
Rating
Weight

DATA DISPLAY	Suppression	2.8

- Temporary suppression of displayed data -1 __ __ .
- Labeling display suppression -2 __ __ .
- Signaling changes to suppressed data -3 __ __ .
- Resuming display of suppressed data -4 __ __ .

DATA DISPLAY	Design Change	2.9

- Flexible design for data display -1 __ __ .

Comment
Rating
Weight

SEQUENCE CONTROL — General 3.0		Weight	Rating	Comment
• Flexible sequence control	-1			
• Minimal user actions	-2			
• Control matched to user skill	-3			
• User initiative in sequence control	-4			
• Control by explicit user action	-5			
• Consistent user actions	-6			
• Logical transaction sequences	-7			
• Distinctive display of control information	-8			
• Displayed context	-9			
• Consistent terminology for sequence control	-10			
• Feedback for control entries	-11			
o Indicating completion of processing	-12			
o Compatibility with user expectations	-13			
• User-paced sequence control	-14			
• Appropriate computer response time	-15			
• Control availability	-16			
o Indicating control lockout	-17			
o Interrupt to end control lockout	-18			
• Control by simultaneous users	-19			

SEQUENCE CONTROL — Dialogue Type 3.1		Weight	Rating	Comment
• Dialogue matched to user and task	-1			
• Appropriate computer response time	-2			

SEQUENCE CONTROL — Question & Answer 3.1.1		Weight	Rating	Comment
• Question-and-answer dialogue	-1			
• Questions displayed singly	-2			
• Recapitulating prior answers	-3			
• Sequence compatible with source documents	-4			

SEQUENCE CONTROL — Form Filling 3.1.2		Weight	Rating	Comment
• Form filling for data entry	-1			
• Form filling for control entry	-2			
• Defaults for control entry	-3			
• Consistent format for control forms	-4			

Comment┐
Rating┐ |
Weight┐ | |
▽ ▽ ▽

SEQUENCE CONTROL	Menu Selection	3.1.3

		Weight	Rating	Comment
• Menu selection	-1	__	__	.
• Single selection per menu	-2	__	__	.
• Single-column list format	-3	__	__	.
• Menu selection by pointing	-4	__	__	.
o Large pointing area for option selection	-5	__	__	.
o Dual activation for pointing	-6	__	__	.
• Menu selection by keyed entry	-7	__	__	.
o Standard area for code entry	-8	__	__	.
• Feedback for menu selection	-9	__	__	.
• Menu options worded as commands	-10	__	__	.
o Option wording consistent with command language	-11	__	__	.
• Letter codes for menu selection	-12	__	__	.
o Consistent coding of menu options	-13	__	__	.
• Explicit option display	-14	__	__	.
o Complete display of menu options	-15	__	__	.
o Menu options dependent on context	-16	__	__	.
• Consistent display of menu options	-17	__	__	.
• Menus distinct from other displayed information	-18	__	__	.
• Logical ordering of menu options	-19	__	__	.
• Logical grouping of menu options	-20	__	__	.
o Logical ordering of grouped options	-21	__	__	.
o Labeling grouped options	-22	__	__	.
• Hierarchic menus for sequential selection	-23	__	__	.
o General menu	-24	__	__	.
o Minimal steps in sequential menu selection	-25	__	__	.
o Easy selection of important options	-26	__	__	.
o Automatic cursor placement	-27	__	__	.
• Indicate current position in menu structure	-28	__	__	.
o Control options distinct from menu branching	-29	__	__	.
o Consistent design of hierarchic menus	-30	__	__	.
o Return to higher-level menus	-31	__	__	.
o Return to general menu	-32	__	__	.
• By-passing menu selection with command entry	-33	__	__	.
o Stacking menu selections	-34	__	__	.

Comment
Rating
Weight

SEQUENCE CONTROL	Function Keys 3.1.4			
• Function keys for critical control entries	-1	__	__	.
o Function keys for frequent control entries	-2	__	__	.
o Function keys for interim control entries	-3	__	__	.
• Distinctive labeling of function keys	-4	__	__	.
o Labeling multifunction keys	-5	__	__	.
• Single keying for frequent functions	-6	__	__	.
o Single activation of function keys	-7	__	__	.
• Feedback for function key activation	-8	__	__	.
• Indicating active function keys	-9	__	__	.
o Disabling unneeded function keys	-10	__	__	.
• Single key for continuous functions	-11	__	__	.
• Consistent assignment of function keys	-12	__	__	.
o Consistent functions in different operational modes	-13	__	__	.
• Return to base-level functions	-14	__	__	.
• Distinctive location	-15	__	__	.
o Layout compatible with use	-16	__	__	.

SEQUENCE CONTROL — Command Language — 3.1.5

		Weight	Rating	Comment
• Command language	-1	___	___	.
• Standard display area for command entry	-2	___	___	.
• Functional command language design	-3	___	___	.
• Layered command language	-4	___	___	.
• Familiar wording	-5	___	___	.
o Consistent wording of commands	-6	___	___	.
o Distinctive wording of commands	-7	___	___	.
• User-assigned command names	-8	___	___	.
• User-requested prompts	-9	___	___	.
o General list of commands	-10	___	___	.
• Command stacking	-11	___	___	.
o User definition of macro commands	-12	___	___	.
• Minimal command punctuation	-13	___	___	.
o Standard command delimiter	-14	___	___	.
• Ignoring blanks in command entry	-15	___	___	.
• Abbreviation of commands	-16	___	___	.
• Standard techniques for command editing	-17	___	___	.
o Interpreting misspelled commands	-18	___	___	.
• Correcting command entry errors	-19	___	___	.
o Aborting erroneous commands	-20	___	___	.
• Reviewing destructive commands	-21	___	___	.

SEQUENCE CONTROL — Query Language — 3.1.6

		Weight	Rating	Comment
• Query language	-1	___	___	.
• Natural organization of data	-2	___	___	.
o Coherent representation of data organization	-3	___	___	.
• Task-oriented wording	-4	___	___	.
• Flexible query formulation	-5	___	___	.
• Minimal need for quantifiers	-6	___	___	.
• Logic to link queries	-7	___	___	.
• Linking sequential queries	-8	___	___	.
• Confirming large-scale retrieval	-9	___	___	.

		Weight	Rating	Comment

SEQUENCE CONTROL	Natural Language	3.1.7
o Constrained natural language	-1	___ ___ .

SEQUENCE CONTROL	Graphic Interaction	3.1.8
• Graphic interaction	-1	___ ___ .
• Menu selection as complementary dialogue	-2	___ ___ .

SEQUENCE CONTROL	Transaction Selection	3.2
• User control in transaction selection	-1	___ ___ .
• General list of control options	-2	___ ___ .
o Organization and labeling of listed options	-3	___ ___ .
• Indicating appropriate control options	-4	___ ___ .
o Prompting control entries	-5	___ ___ .
• Cursor placement for pointing at options	-6	___ ___ .
o Cursor placement for keyed entry of options	-7	___ ___ .
• Displaying option codes	-8	___ ___ .
• Task-oriented wording for options	-9	___ ___ .
• Only available options offered	-10	___ ___ .
• Indicating control defaults	-11	___ ___ .
• Consistent CONTINUE option	-12	___ ___ .
• Stacked commands	-13	___ ___ .
o Consistent order in command stacking	-14	___ ___ .
o Abbreviation in command stacking	-15	___ ___ .
o Minimal punctuation of stacked commands	-16	___ ___ .
o Standard delimiter in command stacking	-17	___ ___ .
• User definition of macro commands	-18	___ ___ .
• User-specified transaction timing	-19	___ ___ .

Comment
Rating
Weight

SEQUENCE CONTROL	Interrupt 3.3	Weight	Rating	Comment
• User interruption of transactions	-1	__	__	.
• Distinctive interrupt options	-2	__	__	.
o CANCEL option	-3	__	__	.
o BACKUP option	-4	__	__	.
o RESTART option	-5	__	__	.
o ABORT option	-6	__	__	.
o END option	-7	__	__	.
o PAUSE/CONTINUE options	-8	__	__	.
o Indicating PAUSE status	-9	__	__	.
o SUSPEND option	-10	__	__	.
o Indicating SUSPEND status	-11	__	__	.

SEQUENCE CONTROL	Context Definition 3.4	Weight	Rating	Comment
• Defining context for users	-1	__	__	.
o Context established by prior entries	-2	__	__	.
o Record of prior entries	-3	__	__	.
• Display of operational mode	-4	__	__	.
• Display of control parameters	-5	__	__	.
• Highlighting selected data	-6	__	__	.
• Consistent display of context information	-7	__	__	.

		Weight	Rating	Comment

SEQUENCE CONTROL — Error Management 3.5

• Appropriate response to all entries	-1	___	___	.
• Command editing	-2	___	___	.
• Prompting command correction	-3	___	___	.
• Errors in stacked commands	-4	___	___	.
o Partial execution of stacked commands	-5	___	___	.
• Explicit entry of corrections	-6	___	___	.
• User confirmation of destructive entries	-7	___	___	.
o User warned of potential data loss	-8	___	___	.
o Distinctive CONFIRM action	-9	___	___	.
• UNDO to reverse control actions	-10	___	___	.
o Preventing data loss at LOG-OFF	-11	___	___	.
o Immediate data correction	-12	___	___	.
o Flexible BACKUP for error correction	-13	___	___	.

SEQUENCE CONTROL — Alarms 3.6

• Alarm definition by user	-1	___	___	.
• Distinctive and consistent alarms	-2	___	___	.
• Alarm acknowledgement	-3	___	___	.
o Alarm reset	-4	___	___	.
o Acknowledgement of critical alarms	-5	___	___	.

SEQUENCE CONTROL — Design Change 3.7

• Flexible design for sequence control	-1	___	___	.

		Weight	Rating	Comment
USER GUIDANCE General	4.0			
• Standard procedures	-1	___	___	.
• Explicit user actions	-2	___	___	.
• Separate LOG-ON procedure	-3	___	___	.
• Display of guidance information	-4	___	___	.
• Only necessary information displayed	-5	___	___	.
• Consistent display format	-6	___	___	.
o Consistent format for user guidance	-7	___	___	.
• Distincitve format for user guidance	-8	___	___	.
o Distinctive cursor	-9	___	___	.
• Clear control labels	-10	___	___	.
• Clear data labels	-11	___	___	.
• Highlighting critical user guidance	-12	___	___	.
• Consistent coding conventions	-13	___	___	.
o Familiar coding conventions	-14	___	___	.
• Consistent wording	-15	___	___	.
o Familiar wording	-16	___	___	.
o Task-oriented wording	-17	___	___	.
o Affirmative statements	-18	___	___	.
o Active voice	-19	___	___	.
o Temporal sequence	-20	___	___	.
o Consistent grammatical structure	-21	___	___	.
• Flexible user guidance	-22	___	___	.
o Easy ways to get guidance	-23	___	___	.

USER GUIDANCE Status Information	4.1			
• Indicating status	-1	___	___	.
• Automatic LOG-ON display	-2	___	___	.
o LOG-ON delay	-3	___	___	.
• Keyboard lock	-4	___	___	.
• Operational mode	-5	___	___	.
• Other users	-6	___	___	.
• System load	-7	___	___	.
• External systems	-8	___	___	.
• Date and time signals	-9	___	___	.
• Alarm settings	-10	___	___	.

Comment
Rating
Weight

USER GUIDANCE	Routine Feedback	4.2	Weight	Rating	Comment
• Consistent feedback		-1	__	__	.
• Fast response		-2	__	__	.
• Feedback for control entries		-3	__	__	.
o Indicating completion of processing		-4	__	__	.
• Feedback for print requests		-5	__	__	.
• Display identification		-6	__	__	.
o Identifying multipage displays		-7	__	__	.
• Indicating operational mode		-8	__	__	.
o Indicating option selection		-9	__	__	.
o Indicating item selection		-10	__	__	.
• Feedback for user interrupt		-11	__	__	.

USER GUIDANCE	Error Feedback	4.3	Weight	Rating	Comment
• Informative error messages		-1	__	__	.
o Specific error messages		-2	__	__	.
o Task-oriented error messages		-3	__	__	.
• Advisory error messages		-4	__	__	.
• Brief error messages		-5	__	__	.
• Neutral wording for error messages		-6	__	__	.
• Multilevel error messages		-7	__	__	.
• Multiple error messages		-8	__	__	.
o Indicating repeated errors		-9	__	__	.
• Non-disruptive error messages		-10	__	__	.
• Appropriate response time for error messages		-11	__	__	.
• Documenting error messages		-12	__	__	.
• Cursor placement following error		-13	__	__	.
• User editing of entry errors		-14	__	__	.
• Cautionary messages		-15	__	__	.
• User confirmation of destructive entries		-16	__	__	.
• Alarm coding		-17	__	__	.

		Weight	Rating	Comment
USER GUIDANCE Job Aids	4.4			
• Guidance information always available	-1	__	__	.
• General list of control options	-2	__	__	.
• Logical menu structure	-3	__	__	.
• Hierarchic menus	-4	__	__	.
• Guidance for sequence control	-5	__	__	.
o Transaction specific option display	-6	__	__	.
• Prompting entries	-7	__	__	.
o Standard display location for prompting	-8	__	__	.
o User-requested prompts	-9	__	__	.
• Displayed context	-10	__	__	.
• Cues for prompting data entry	-11	__	__	.
• Consistent cursor positioning	-12	__	__	.
• On-line system guidance	-13	__	__	.
o Index of data	-14	__	__	.
o Index of commands	-15	__	__	.
o Dictionary of abbreviations	-16	__	__	.
• Definition of display codes	-17	__	__	.
• Record of past transactions	-18	__	__	.
• HELP	-19	__	__	.
o Standard action to request HELP	-20	__	__	.
o Task-oriented HELP	-21	__	__	.
o Clarifying HELP requests	-22	__	__	.
o Multilevel HELP	-23	__	__	.
o Browsing HELP	-24	__	__	.
• On-line training	-25	__	__	.
o Flexible training	-26	__	__	.
o Adaptive training	-27	__	__	.

Comment
Rating
Weight

USER GUIDANCE	User Records	4.5	Weight	Rating	Comment
• User performance measurement		-1			
• Notifying users		-2			
• Transaction records		-3			
• Data access records		-4			
• Program use records		-5			
• Error records		-6			
• HELP records		-7			

USER GUIDANCE	Design Change	4.6	Weight	Rating	Comment
• Flexible design for user guidance		-1			
• Notifying users of design changes		-2			

Comment
Rating
Weight

DATA TRANSMISSION	General	5.0	Weight	Rating	Comment
• Consistent procedures		-1	___	___	.
• Minimal user actions		-2	___	___	.
• Minimal memory load on user		-3	___	___	.
• Message composition compatible with data entry		-4	___	___	.
o Message viewing compatible with data display		-5	___	___	.
• Flexible user control		-6	___	___	.
• Control by explicit user action		-7	___	___	.

DATA TRANSMISSION	Data Type	5.1	Weight	Rating	Comment
• User-designed formats		-1	___	___	.
• Automatic text formatting		-2	___	___	.
• Unformatted text		-3	___	___	.
• Data forms		-4	___	___	.
o Tables and graphics		-5	___	___	.
• Message highlighting		-6	___	___	.

DATA TRANSMISSION	Sending	5.2	Weight	Rating	Comment
• Source selection		-1	___	___	.
• Destination selection		-2	___	___	.
• Status information		-3	___	___	.
• Assignment of priority		-4	___	___	.
• Message printing		-5	___	___	.

DATA TRANSMISSION	Receiving	5.3	Weight	Rating	Comment
• Source selection		-1	___	___	.
• Destination selection		-2	___	___	.
o Receipt by priority		-3	___	___	.

Comment
Rating
Weight

DATA TRANSMISSION	Transmission Control 5.4	Weight	Rating	Comment
• Functional wording	-1	__	__	.
• Flexible data specification	-2	__	__	.
• Automatic message formatting	-3	__	__	.
• Automatic message routing	-4	__	__	.
• Automatic message initiation	-5	__	__	.
• User review of transmitted data	-6	__	__	.

DATA TRANSMISSION	Feedback 5.5	Weight	Rating	Comment
• Automatic feedback	-1	__	__	.
• Feedback for message sent	-2	__	__	.
• Information about messages received	-3	__	__	.
• User specification of feedback	-4	__	__	.

DATA TRANSMISSION	Queuing 5.6	Weight	Rating	Comment
• Automatic queuing	-1	__	__	.
• Deferring message transmission	-2	__	__	.
o Queuing failed transmissions	-3	__	__	.
• Queuing messages received	-4	__	__	.
• Non-disruptive message receipt	-5	__	__	.
o Priority indicating for messages received	-6	__	__	.
• User review of messages received	-7	__	__	.

DATA TRANSMISSION	Record Keeping 5.7	Weight	Rating	Comment
• Automatic record keeping	-1	__	__	.

DATA TRANSMISSION	Design Change 5.8	Weight	Rating	Comment
• Flexible design for data transmission	-1	__	__	.

Comment
Rating
Weight

DATA PROTECTION	General	6.0	Weight	Rating	Comment
• Automatic security measures		-1	___	___	.
• Consistent procedures		-2	___	___	.
• Control by explicit user action		-3	___	___	.
• Feedback for mode selection		-4	___	___	.
• Appropriate response to all entries		-5	___	___	.
• User review and editing of entries		-6	___	___	.
• Resolving ambiguous entries		-7	___	___	.
• Users warned of threats to security		-8	___	___	.

DATA PROTECTION	User Identification	6.1	Weight	Rating	Comment
• Easy LOG-ON		-1	___	___	.
o Prompting LOG-ON		-2	___	___	.
• User choice of passwords		-3	___	___	.
o Private entry of passwords		-4	___	___	.
• Continuous recognition of user identity		-5	___	___	.

DATA PROTECTION	Data Access	6.2	Weight	Rating	Comment
• Single authorization for data access		-1	___	___	.
• Displayed security classification		-2	___	___	.
• User editing of displayed data		-3	___	___	.
• Protecting displayed data		-4	___	___	.
o Protecting display formats		-5	___	___	.
• Easy display suppression		-6	___	___	.
• Printing protected data		-7	___	___	.
• Automatic records of data access		-8	___	___	.

				Weight	Rating	Comment
DATA PROTECTION	Data Entry/Change	6.3				
•	Single authorization for data entry/change	-1		___	___	.
•	Data entry/change transaction records	-2		___	___	.
•	Protection from data change	-3		___	___	.
•	Segregating real from simulated data	-4		___	___	.
•	Emphasizing accuracy	-5		___	___	.
•	Simple procedures	-6		___	___	.
o	Displaying default values	-7		___	___	.
•	Explicit user actions	-8		___	___	.
o	Single entry of related data	-9		___	___	.
•	User editing of data before entry	-10		___	___	.
o	Immediate error correction	-11		___	___	.
o	User editing of entry errors	-12		___	___	.
•	Flexible BACKUP for error correction	-13		___	___	.
•	Explicit entry of corrections	-14		___	___	.
•	Data verification by user review	-15		___	___	.
o	Automatic data generation	-16		___	___	.
•	Validation of changed data	-17		___	___	.
o	Cross validation of changed data	-18		___	___	.
•	Displaying data to be changed	-19		___	___	.
•	User confirmation of destructive actions	-20		___	___	.

				Weight	Rating	Comment
DATA PROTECTION	Data Transmission	6.4				
•	Automatic protection of transmitted data	-1		___	___	.
•	User review of data before transmission	-2		___	___	.
•	Saving sent data until receipt is confirmed	-3		___	___	.
•	Queuing received messages	-4		___	___	.
•	Printing messages	-5		___	___	.

Comment
Rating
Weight

DATA PROTECTION	Loss Prevention	6.5	Weight	Rating	Comment
• Protecting data from computer failure		-1	__	__	.
• Protecting data from other users		-2	__	__	.
• Protecting data from interrupt actions		-3	__	__	.
• Segregating real from simulated data		-4	__	__	.
• Appropriate ease/difficulty of user actions		-5	__	__	.
• Standard procedures		-6	__	__	.
• Disabling unneeded controls		-7	__	__	.
o Protecting controls		-8	__	__	.
• Data entry independent of cursor placement		-9	__	__	.
• Displaying data to be changed		-10	__	__	.
• User review of interpreted commands		-11	__	__	.
• Protective defaults		-12	__	__	.
• Explicit user action to select destructive modes		-13	__	__	.
• Protecting data from user error		-14	__	__	.
• Distinctive file names		-15	__	__	.
• Preventing data loss at LOG-OFF		-16	__	__	.
• Warning users of potential data loss		-17	__	__	.
• User confirmation of destructive actions		-18	__	__	.
o Distinctive CONFIRM action		-19	__	__	.
• Reversible control actions: UNDO		-20	__	__	.

DATA PROTECTION	Design Change	6.6	Weight	Rating	Comment
• Flexible design for data protection		-1	__	__	.
• Protection from design change		-2	__	__	.

APPENDIX B

SAMPLE CHECKLIST ENTRIES

This appendix presents a portion of the USI design evaluation checklist showing sample entries for data entry functions. In the sample shown here, numbers have been entered from zero to ten to represent agreed weightings and judged ratings of different design guidelines. Those numbers are fanciful, and are included here only for illustrative purposes. They do not represent a design evaluation of any real system.

DESIGN EVALUATION FOR USI SOFTWARE

System: COMCON
Date: 8-15-84
Evaluator: St. Smith

DATA ENTRY — General 1.0	Guideline Number	Weight	Rating	Comment
• Entry via primary display	-1	7	10	.
• Feedback during data entry	-2	10	8	✓
o Fast response	-3	10	10	.
• Single method for entering data	-4	7	7	.
• Defined display areas for data entry	-5	7	10	.
• Consistent method for data change	-6	3	5	✓
• User-paced data entry	-7	0	—	.
• Explicit ENTER action	-8	7	10	.
o ENTER key labeling	-9	3	10	.
• Explicit CANCEL action	-10	0	—	.
• Feedback for completion of data entry	-11	7	8	.
o Feedback for repetitive data entries	-12	0	—	.
o Feedback when changing data	-13	7	10	.
• Keeping data items short	-14	0	—	.
o Partitioning long data items	-15	0	—	.
• Optional abbreviation	-16	3	5	✓
o Distinctive abbreviation	-17	3	10	.
o Consistent abbreviation rule	-18	5	5	✓
o Minimal exceptions to abbreviation rule	-19	3	10	.
o Minimal deviation from abbreviation rule	-20	3	5	✓
o Fixed abbreviation length	-21	0	—	.
o Clarifying unrecognized abbreviations	-22	5	10	.
• Prompting data entry	-23	5	5	✓
• Character entry via single keystroke	-24	0	—	.
o Minimal shift keying	-25	0	—	.
• Upper/lower case equivalent	-26	0	—	.
• Decimal point optional	-27	0	—	.
• Leading zeros optional	-28	0	—	.
• Single/multiple blanks equivalent	-29	0	—	.

DATA ENTRY	Position Designation	1.1	Weight	Rating	Comment
• Distinctive cursor		-1	10	10	.
o Non-obscuring cursor		-2	10	7	✓
o Precise pointing		-3	0	—	.
• Explicit activation		-4	10	10	.
o Fast Response		-5	10	3	✓
• Stable cursor		-6	3	10	.
• Responsive cursor control		-7	0	—	.
• Consistent incremental positioning		-8	10	10	.
o Variable step size		-9	0	—	.
o Proportional spacing		-10	0	—	.
• Continuous cursor positioning		-11	0	—	.
• Direct pointing		-12	10	10	.
o Large pointing area for option selection		-13	7	9	✓
• Cursor control at keyboard		-14	5	10	✓
• Compatible control of cursor movement		-15	0	—	.
• Minimal use of multiple cursors		-16	0	—	.
o Distinctive multiple cursors		-17	0	—	.
o Distinctive control of multiple cursors		-18	0	—	.
o Compatible control of multiple cursors		-19	0	—	.
• Consistent HOME position		-20	3	10	.
• Consistent cursor placement		-21	5	5	✓
• Easy cursor movement to data fields		-22	3	7	✓
• Display format protection		-23	7	10	.
• Data entry independent of cursor placement		-24	7	10	.

DATA ENTRY	Direction Designation	1.2	Weight	Rating	Comment
• Analog entry of estimated direction		-1	0	—	.
• Keyed entry of quantified direction		-2	0	—	.

		Weight	Rating	Comment
DATA ENTRY	**Text 1.3**			
• Adequate display capacity	-1	10	10	.
• Full editing capabilities during text entry	-2	5	3	.
• Free cursor movement	-3	5	10	.
o Control entries distinct from text	-4	5	10	.
• Natural units of text	-5	8	5	✓
o Control entry based on units of text	-6	3	2	✓
o Highlighting specified text	-7	3	10	.
o Cursor movement by units of text	-8	8	5	✓
• String search	-9	0	—	.
o Upper/lower case equivalent in search	-10	0	—	.
o Specifying case in search	-11	0	—	.
o Global search and replace	-12	0	—	.
o Case in global search and replace	-13	0	—	.
• Automatic pagination aids	-14	0	—	.
o User control of pagination	-15	0	—	.
o Controlling integrity of text units	-16	0	—	.
• Automatic line break	-17	8	10	.
o Consistent word spacing	-18	8	10	.
o Hyphenation by users	-19	5	10	.
• Format control by user	-20	10	8	.
• Establishing predefined formats	-21	10	10	.
o Storing user-defined formats	-22	5	7	.
• Moving text	-23	1	0	.
• Storing frequently used text	-24	10	10	.
• Necessary data displayed	-25	10	8	✓
o Text distinct from annotation	-26	5	5	.
• Printing for proofreading	-27	0	—	.
• Text displayed as printed	-28	0	—	.
• Flexible printing options	-29	0	—	.
• Information on printing status	-30	0	—	.
• Auditory signals for alerting users	-31	8	5	✓
• Protecting text during page overruns	-32	5	10	.
• Confirming actions in DELETE mode	-33	10	10	.
• Reversible actions	-34	6	10	.
• User confirmation of editing changes	-35	3	0	.

		Weight	Rating	Comment

DATA ENTRY — Data Forms 1.4

		Weight	Rating	Comment
• Single entry of related data	-1	10	10	.
• Flexible interrupt	-2	8	10	.
• Minimal use of delimiters	-3	0	—	.
o Standard delimiter character	-4	0	—	.
• Data field labels	-5	10	2	✓
o Consistent labeling	-6	10	8	✓
o Protected labels	-7	7	10	.
o Labels close to data fields	-8	3	10	.
• Marking field boundaries	-9	6	8	✓
o Prompting field length	-10	3	10	.
o Marking required/optional data fields	-11	0	—	.
• Automatic justification of variable-length entries	-12	0	—	.
• Explicit tabbing to data fields	-13	8	10	.
• Distinctive label format	-14	9	8	✓
o Consistent label format	-15	7	8	✓
o Label punctuation as entry cue	-16	0	—	.
• Informative labels	-17	6	8	✓
• Data format cueing in labels	-18	0	—	.
o Labeling units of measurement	-19	0	—	.
o Familiar units of measurement	-20	0	—	.
o Alternative units of measurement	-21	0	—	.
• Form compatible for data entry and display	-22	10	10	.
o Form compatible with source documents	-23	0	—	.
• Minimal Cursor positioning	-24	6	10	.
o Data items in logical order	-25	10	8	✓
• Automatic cursor placement	-26	3	5	✓

DATA ENTRY — Tables 1.5

		Weight	Rating	Comment
• Tables for related data sets	-1	10	10	.
• Distinctive labels	-2	2	10	.
o Informative labels	-3	4	10	.
• Tabbing within rows	-4	10	10	.
o Tabbing within columns	-5	10	10	.
• Automatic justification of entries	-6	0	—	.
o Justification of numeric entries	-7	0	—	.
• Aiding entry of duplicative data	-8	0	—	.
• Row scanning cues	-9	8	10	.

Comment
Rating
Weight

DATA ENTRY — Graphics 1.6

(No entries)

DATA ENTRY — Data Validation 1.7

		Weight	Rating	Comment
• Automatic data validation	-1	10	10	.
• Accepting correct entries	-2	10	8	✓
• Non-disruptive error messages	-3	4	5	.
• Deferral of required data entry	-4	0	—	✓
o Reminder of deferred entry	-5	0	—	.
• Timely validation of sequential transactions	-6	0	—	.
• Optional item-by-item validation	-7	0	—	.

DATA ENTRY — Other Data Processing 1.8

		Weight	Rating	Comment
• Default values	-1	0	—	.
o User definition of default values	-2	0	—	.
o Display of default values	-3	0	—	.
o Easy confirmation to enter default values	-4	0	—	.
o Temporary replacement of default values	-5	0	—	.
• Automatic generation of routine data	-6	10	10	.
• Automatic computation of derived data	-7	10	8	✓
• User review of prior entries	-8	0	—	.
• Automatic entry of redundant data	-9	0	—	.
• Automatic cross-file updating	-10	10	10	✓

DATA ENTRY — Design Change 1.9

		Weight	Rating	Comment
• Flexible design for data entry	-1	10	2	✓

APPENDIX C

SAMPLE DESIGN ANALYSIS

The following pages illustrate the kind of analysis that could be based on the USI design evaluation checklist. The example shown here assumes that the sample checklist for data entry functions invented in Appendix B has been extended to include all USI functional areas. Since that checklist represents a fictional evaluation, this design analysis is also fictional, and does not represent an evaluation of any actual system.

In this sample analysis, it is assumed that an evaluator has assigned ratings to all required items in the checklist. This is indicated in the computer-generated report. In an actual system evaluation, that will not always be true. Some items may have been left unrated because of inadequate design information. In such a case, the computer might provide a partial design analysis. Or, if many ratings have been omitted, the computer might report that no useful analysis can be made.

The analysis begins at a general level, assessing overall USI design quality in relation to defined requirements. The analysis continues at levels of increasing detail, to provide a diagnostic evaluation of design compliance for different USI functional areas, and for specific functions within each area.

Various information is provided at different levels of analysis, including derived measures of design quality. The measures proposed here illustrate several promising possibilities. Still other measures might be derived from the itemized design evaluation checklist to match particular system requirements.

As noted in the text of this report, such a detailed design analysis can be tedious using manual methods, although it is certainly feasible. It would be helpful to provide computer aids for such an analysis. If the weightings and ratings of USI design evaluation can be assigned on-line, as entries to an automated checklist, then it should be possible to provide a computer-generated summary and detailed diagnostic information, quickly and accurately. The computer output might look much like that imagined in the following pages.

DESIGN EVALUATION FOR USI SOFTWARE

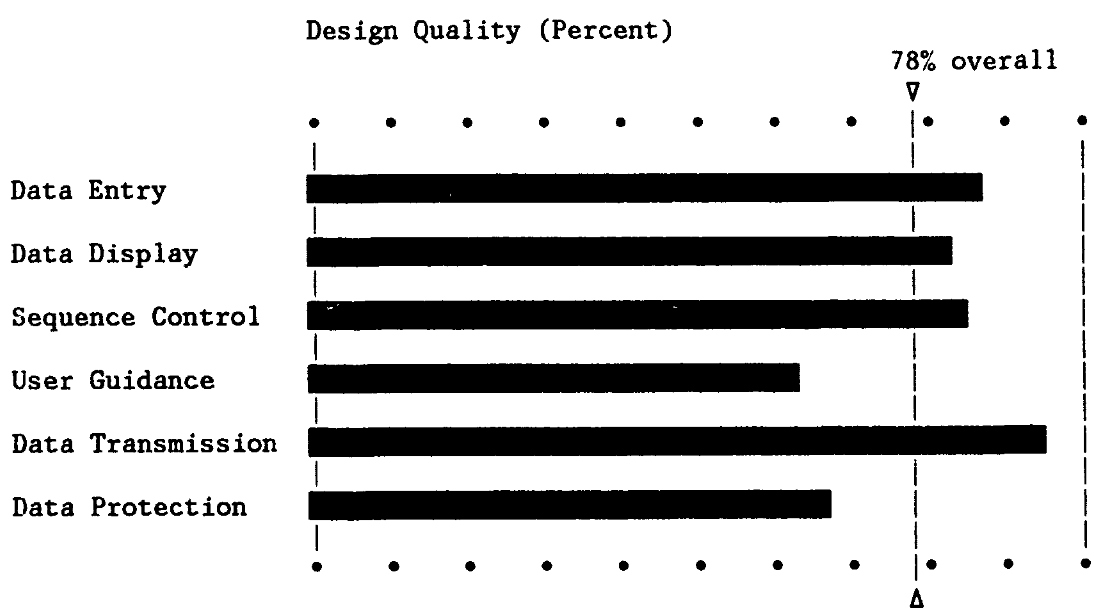
System: COMCON (proposed design)
Date: 15 August 1984
Evaluator: S. L. Smith

SUMMARY

All required items have been rated, and so a comprehensive analysis of user interface software can be made.

Overall achievement of USI design objectives is here estimated at 78 percent. This figure represents a weighted average of assigned ratings for all required items, in relation to maximum possible ratings.

This figure indicates generally good achievement of design objectives, but with a need for further improvement. A profile of weighted design quality in six USI functional areas shows that improvement is needed in the design of user guidance and data protection functions:



A more detailed evaluation of USI design is presented in the following pages.

FUNCTIONAL AREAS

Mean weights and ratings of design features for six USI functional areas are tabulated below.

	(A)	(B)	(C)	(D)	(E)	(F)	(G)
	Design Guidelines			Required Guidelines		Computed Design Quality	
	Listed	Required No.	%	Mean Weight	Mean Rating	Value	%
1. Data Entry	143	83	58	6.9	8.3	4846	85
2. Data Display	163	138	85	7.8	7.9	8678	81
3. Sequence Control	168	49	29	7.3	8.6	3010	84
4. User Guidance	97	39	40	8.1	6.4	1977	62
5. Data Transmission	40	16	40	7.7	8.9	1145	93
6. Data Protection	68	56	82	6.3	6.9	2358	66
Overall	679	381	56	7.3	7.8	22014	78

(A) = total number of guidelines in the design evaluation checklist.

(B) = number of required guidelines, i.e., those weighted above zero.

(C) = percent of listed guidelines that are required. [= 100 x B/A]

(D) = mean weight (1-10) assigned to required guidelines.

(E) = mean rating (0-10) of design compliance for required guidelines.

(F) = computed sum of weighted design compliance, i.e., total of weights multiplied by ratings for required guidelines.

(G) = percent of actual design quality achieved in relation to maximum potential compliance.
[= 100 x F/(BxDx10) subject to rounding errors]

Overall design bias computed from this table is 1.5 percent. Design bias refers to a tendency for compliance to be greater for required design features that are assigned greater weight. Design bias is computed as 100 x (F/(BxDxE) - 1).

The computed bias indicates that design compliance for this system is not significantly related to the weighting of required design features. For further improvement of USI design, it will be necessary to examine instances in which high-weighted features have received low ratings of design compliance.

A more detailed evaluation is presented separately for each USI functional area in the following pages.

DATA ENTRY

Mean weights and ratings of design guidelines for data entry functions are tabulated below. For a more detailed review of data entry functions, consult items in that section of the USI design evaluation checklist.

	(A)	(B)	(C)	(D)	(E)	(F)	(G)
	Design Guidelines			Required Guidelines		Computed Design Quality	
	Listed	Required No.	%	Mean Weight	Mean Rating	Value	%
1.0 General	29	17	59	5.6	8.2	814	87
1.1 Position	24	14	58	6.8	8.9	882	88
1.2 Direction	2	0	0	-	-	-	-
1.3 Text	35	23	66	6.6	7.3	1211	80
1.4 Data Forms	26	16	62	7.3	8.4	969	84
1.5 Tables	9	6	67	7.3	10.0	440	100
1.6 Graphics	0	-	-	-	-	-	-
1.7 Data Validation	7	3	43	8.0	9.7	230	96
1.8 Other Processing	10	3	30	10.0	9.3	280	93
1.9 Design Change	1	1	100	10.0	2.0	20	20
Overall Data Entry	143	83	58	6.9	8.3	4846	85

Evaluator comments on specific guidelines are provided for 31 items.

The following high-weighted guidelines with low ratings of design compliance will require special attention to improve USI design.

		Weight	Rating
1.1-5	Fast response	10	5
1.4-5	Data field labels	10	2
1.9-1	Flexible design for data entry	10	2

* * * * *

[A complete design analysis would continue with similar pages for five other USI functional areas. Those pages are omitted in this sample.]